**DataCheck.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.OleDb;

using ESRI.ArcGISExplorer;

using ESRI.ArcGISExplorer.Application;

using ESRI.ArcGISExplorer.Mapping;

using ESRI.ArcGISExplorer.Geometry;

using ESRI.ArcGISExplorer.Data;

using ESRI.ArcGISExplorer.Threading;

using DataChecker;

/\*

DataChecker Extension

Author: Adon Moskal

April 2010 - June 2010

\*/

namespace DataChecker

{

public class DataCheck : ESRI.ArcGISExplorer.Application.Extension

{

public override void OnStartup()

{

Form1 form = new Form1();

form.Show();

form.Activate();

}

public override void OnShutdown()

{

}

}

}

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.OleDb;

using ESRI.ArcGISExplorer;

using ESRI.ArcGISExplorer.Application;

using ESRI.ArcGISExplorer.Mapping;

using ESRI.ArcGISExplorer.Geometry;

using ESRI.ArcGISExplorer.Data;

using ESRI.ArcGISExplorer.Threading;

namespace DataChecker

{

public partial class Form1 : Form

{

//data members

MapDisplay md = ESRI.ArcGISExplorer.Application.Application.ActiveMapDisplay;

TYear currYear;

TEngine currEngine;

Boolean mismatch;

System.IO.StreamWriter myStreamWriter;

string path;

public const int STARTYEAR = 1846;

public const int ENDYEAR = 2009;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

currEngine = new TEngine();

path = currEngine.getPath();

//create the log

myStreamWriter = new System.IO.StreamWriter(path + "Data Checker Log.txt");

for (int i = STARTYEAR; i < ENDYEAR; i++)

{

mismatch = false;

//if a kmz model file exists, check the data for that year

if (System.IO.File.Exists(path + "Output\\" + i.ToString() + "\\" + i.ToString() + ".kmz"))

{

currYear = new TYear();

currYear.loadData(path, i.ToString());

//if the Access DB has no data, something is wrong and the error should be triggered

if (currYear.Dv.Table.Rows.Count == 0)

mismatch = true;

else

//for each row in the Access DB, compare to the shapefile - in this case, only the name field

//is being checked

for (int k = 0; k < currYear.Dv.Table.Rows.Count; k++)

{

DataRow DR = currYear.Dv.Table.Rows[k];

Row R = currYear.Shape.Table.GetRow(k);

if (DR[4].ToString() != R.Values[4].ToString())

mismatch = true;

}

//if data mismatch occurs, alert the user as to the year in question

if ((mismatch == true) || (currYear.Dv.Table.Rows.Count == 0))

{

MessageBox.Show("Error! Data Mismatch" + Environment.NewLine

+ "Please re-export year: " + i.ToString());

//write the year to the log along with the date and time

myStreamWriter.WriteLine(DateTime.Now + " " + i.ToString());

}

}

}

//close the log streamwriter

myStreamWriter.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

}

private void Form1\_Shown(object sender, EventArgs e)

{

//close the datachecker automatically

this.Close();

}

}

}

**TEngine.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.OleDb;

using ESRI.ArcGISExplorer;

using ESRI.ArcGISExplorer.Application;

using ESRI.ArcGISExplorer.Mapping;

using ESRI.ArcGISExplorer.Geometry;

using ESRI.ArcGISExplorer.Data;

using ESRI.ArcGISExplorer.Threading;

namespace DataChecker

{

public class TEngine

{

//data members

string path;

string newpath;

//constructor

public TEngine()

{

}

//gets the path of the data for the data checker to read

public string getPath()

{

MessageBox.Show("Please select a file to specify a directory for the data checker extension.");

OpenFileDialog ofd = new OpenFileDialog();

if (ofd.ShowDialog() == DialogResult.OK)

{

path = System.IO.Path.GetDirectoryName(ofd.FileName);

}

MessageBox.Show("Working Directory: \"" + path + "\"", "Directory");

//replace escape characters so the string can be used as a path

newpath = path.Replace("\\", "\\\\");

newpath = newpath + "\\\\";

return newpath;

}

}

}

**TYear.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.OleDb;

using ESRI.ArcGISExplorer;

using ESRI.ArcGISExplorer.Application;

using ESRI.ArcGISExplorer.Mapping;

using ESRI.ArcGISExplorer.Geometry;

using ESRI.ArcGISExplorer.Data;

using ESRI.ArcGISExplorer.Threading;

namespace DataChecker

{

public class TYear

{

//data members

protected FeatureLayer shape;

public FeatureLayer Shape

{

get { return shape; }

set { shape = value; }

}

protected DataView dv;

public DataView Dv

{

get { return dv; }

set { dv = value; }

}

protected DataTable dt;

//constructor

public TYear()

{

}

//load the data to check

public void loadData(string newpath, string year)

{

try

{

//load data from the Access DB

OleDbConnection con = new OleDbConnection(@"Provider=Microsoft.JET.OLEDB.4.0;"

+ @"data source=" + newpath + "eHeritage GeoDatabase.mdb");

con.Open();

DataSet ds = new DataSet();

OleDbDataAdapter adapter = new OleDbDataAdapter("Select \* from buildings\_" + year, con);

adapter.Fill(ds);

con.Close();

dt = ds.Tables[0];

dv = dt.DefaultView;

//load the shapefile

shape = FeatureLayer.OpenShapefile(newpath + "Output\\" + year + "\\" + year + ".shp");

}

catch (ConnectionException)

{

MessageBox.Show("Connection Error");

}

}

}

}